Fast Multi-scale Algorithms for Representation and Analysis of Data

Linda Ness (Telcordia)
Joint Work with Devasis Bassu (Telcordia),
Peter Jones (Yale), and Vladimir Rokhlin (Yale)

FERC Technical Conference on Increasing Real-Time and Day-Ahead Market Efficiency through Improved Software

June 30, 2011

Overview

- Suite of fast, scalable algorithms for:
  - Normalized multi-scale representation
    - Of windows of streaming data
    - Representations may be high-dimensional
  - Computation of low-dimensional models
    - Algorithms enable fast update of models from streaming data
- Potential applications
  - Characterization of operational regimes
  - Timely detection of (subtle) regime changes
  - Simulation/analysis using synthesized multi-scale data
  - Improved prediction based on real-time models
  - Improved prediction for renewable energy sources

Outline of the Talk

- Overview of technical approach
  - The Product Formula Representation
  - Diffusion Mapping Representation
  - Fast Randomized Algorithms (RSVD, RANN)
- Data experiments demonstrating applicability
- Planning for new demonstrations

Product Formula Representation

Some Haar-like functions

"The Theory of Weights and the Dirichlet Problem for Elliptic Equations" by R. Fefferman, C. Kenig, and J.Pipher (Annals of Math., 1991). We first define the "L^∞ normalized Haar function" h_I for an interval I of form $[j2^{-n},(j+1)2^{-n}]$ to be of form

$$h_I = +1 \text{ on } [j2^{-n},(j+1/2)2^{-n})$$

and

$$h_I = -1 \text{ on } [(j+1/2)2^{-n},(j+1)2^{-n}).$$

The only exception to this rule is if the right hand endpoint of I is 1. Then we define

$$h_I(1) = -1.$$

The Product Formula

- **Theorem (F,K,P)**: A Borel probability measure μ on [0,1] has a unique representation as

$$\prod(1 + a_I h_I) ,$$

where the coefficients a_I are $\in$ [-1,+1]. Conversely, if we choose any sequence of coefficients $a_I \in$ [-1,+1], the resulting product is a Borel probability measure on [0,1].

Note: For general positive measures, just multiply by a constant. Similar result on $[0,1]^d$.

Note: See "The Theory of Weights and the Dirichlet Problem for Elliptic Equations" by R. Fefferman, C. Kenig, and J.Pipher (Annals of Math., 1991)

Relative "Volume"

The coefficients a_I are computed simply by computing relative measure ("volume") on the two halves of each interval I.
Let L and R = left (resp. right) halves of I.
Solve:

$$\mu(L) = \tfrac{1}{2}(1 + a_I)\,\mu(I)$$

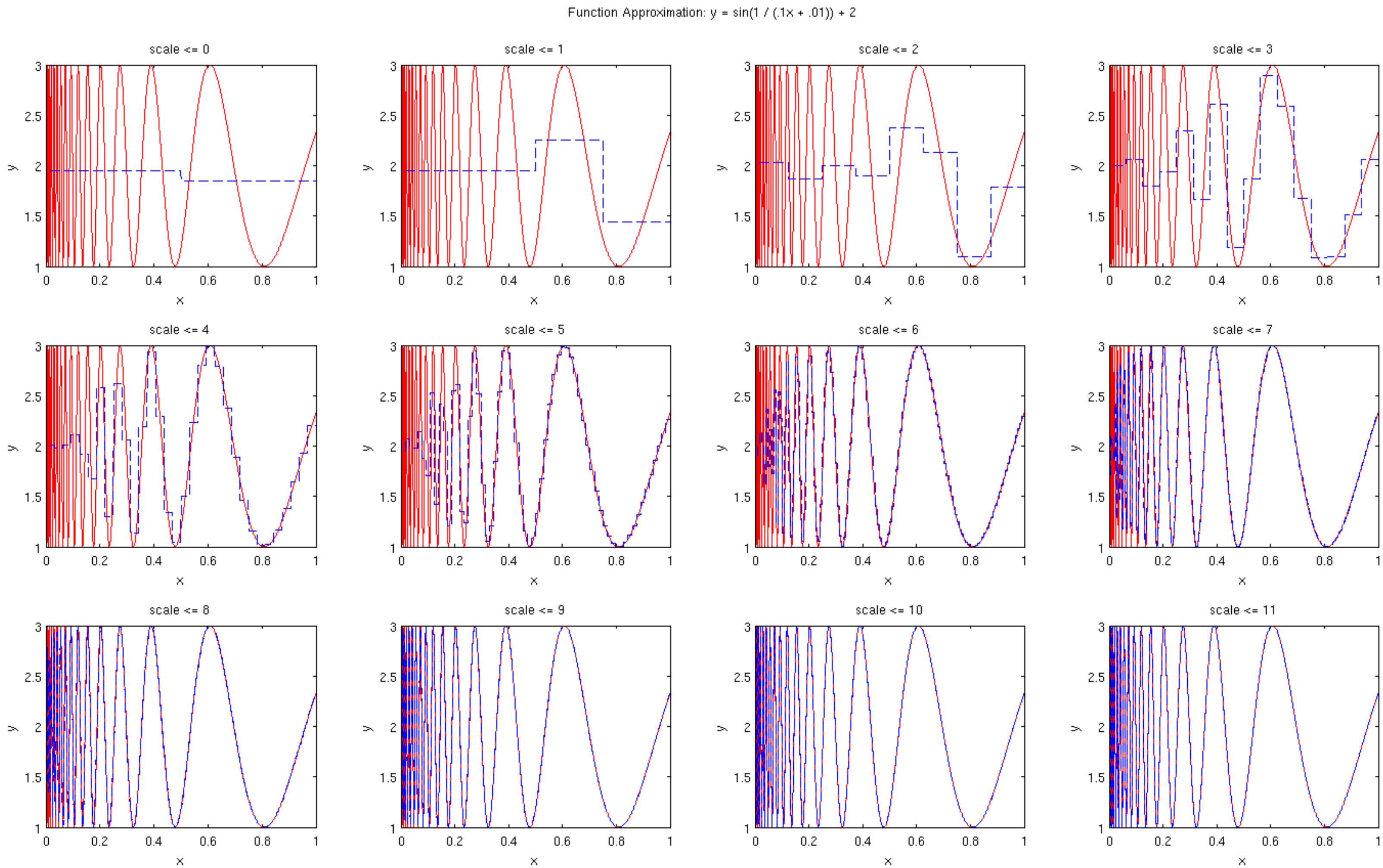
$$\mu(R) = \tfrac{1}{2}(1 - a_I)\,\mu(I)$$

Then $-1 \leq a_I \leq +1$ because μ is nonnegative.

Why use it Instead of Wavelets?

- The coefficients measure relative measure instead of measure. All scales and locations are counted as “equal”.
- Allows another method to represent the signal, where one immediately detects large changes in relative volume. (Anomaly detection)
- Multiple channels are immediately normalized if one looks just at the coefficients instead of absolute volume.
- One cannot easily synthesize using Haar wavelets and the “usual” expansion” (How to keep the function positive?).

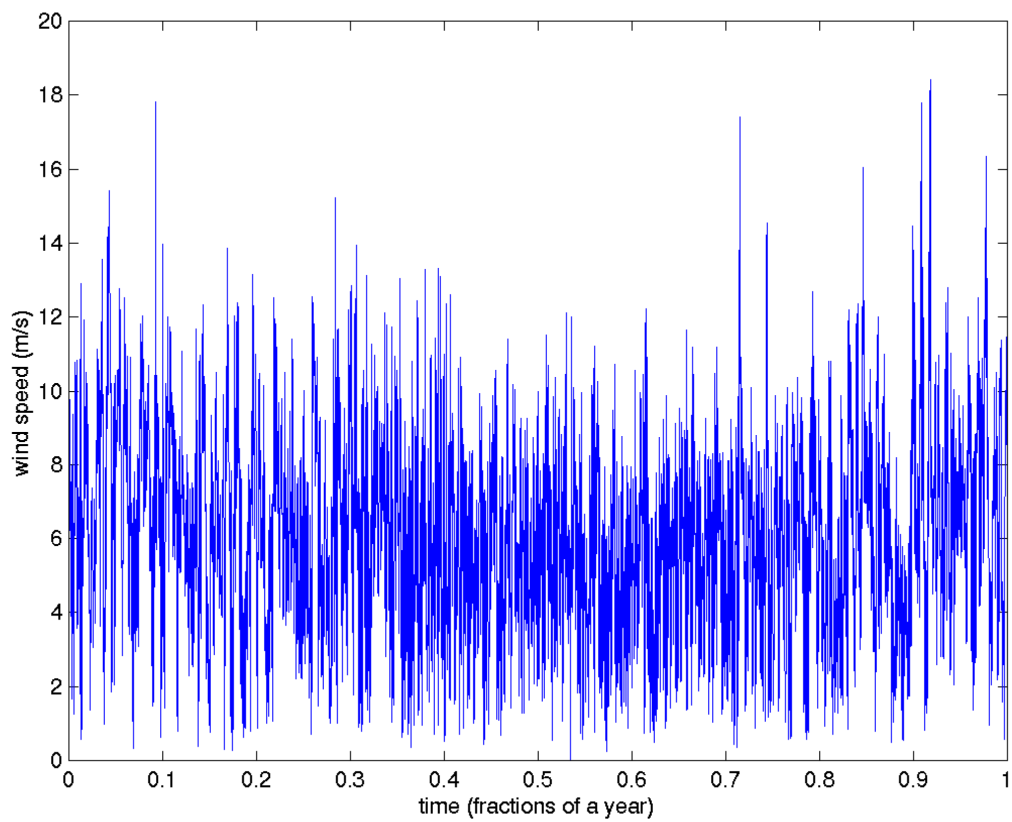
Product Representation: Example



Function Approximation Scale 0 – 11: $y = \sin(1/(x + .01)) + 2$

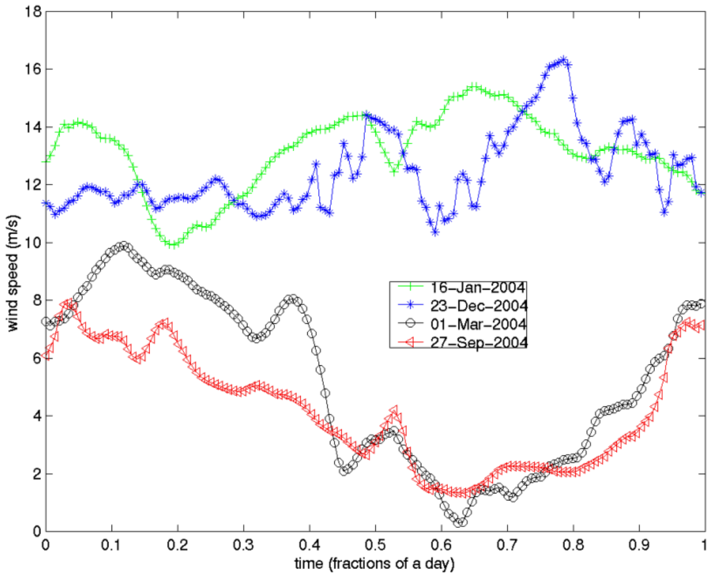
Wind Data -

- Dataset from NREL gives wind speeds and potential wind turbine power levels for a large number of locations and elevations across the U.S., every 10 minutes for three years
  - “These wind power data ("Data") are provided by the National Renewable Energy Laboratory ("NREL"), which is operated by the Alliance for Sustainable Energy ("Alliance") for the U.S. Department Of Energy ("DOE"). “
- We looked at wind speeds for a single year, single location and elevation.
- We can compare the product coefficients for each day with the original time series.
- **Product coefficients provide normalized representations of multi-scale wind patterns**

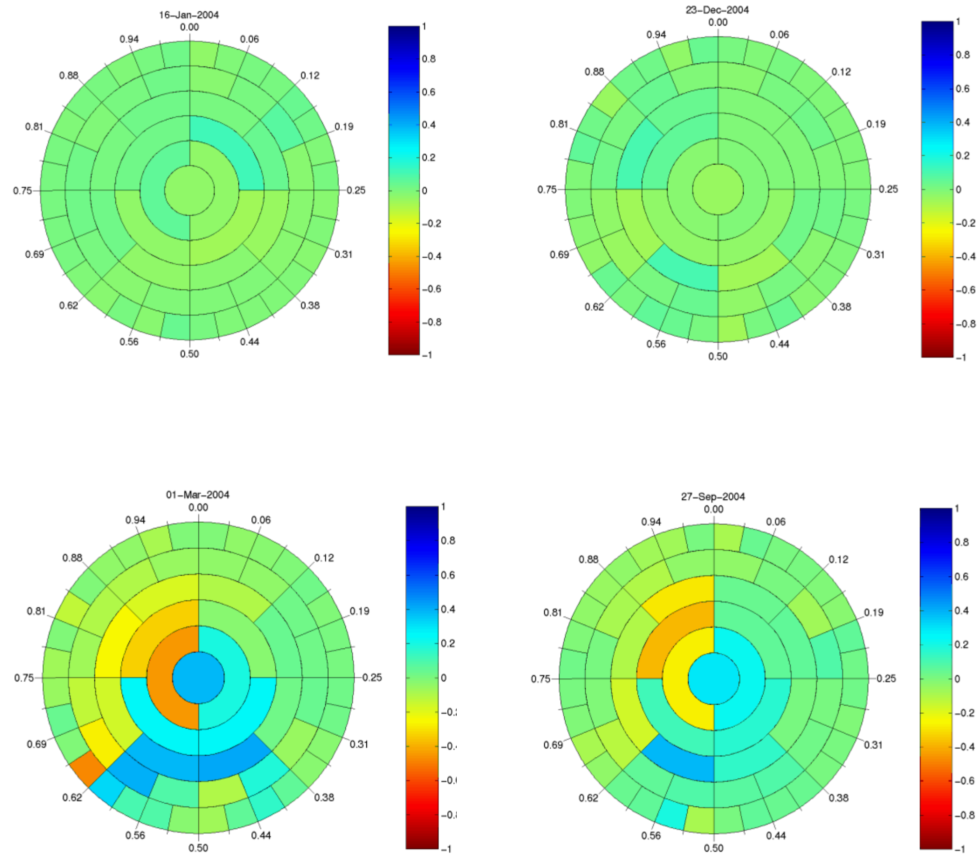


Wind Analysis by David Shallcross (Telcordia)
Fast Multi-scale Algorithms for Representation and Analysis – BNJR - FERC Technical Conference, June 2011.

Four Specific Days of Wind –



Product coefficients specify multi-scale patterns



•The Jan 16 and Dec 23 wind patterns have relatively little variation, so product coefficients are small

•The Mar 1 and Sep 27 wind patterns have minimum in early afternoon,

- •so the scale 0 coefficients are > 0,
- •the first scale 1 coefficient is > 0,
- •second scale 1 coefficient is < 0.

Wind Analysis by David Shallcross (Telcordia)

Fast Multi-scale Algorithms for Representation and Analysis – BNJR - FERC Technical Conference, June 2011.

Diffusion Mapping Representation

Diffusion Geometry

The idea behind Diffusion Geometry is to put a new, nonlinearly defined representation af the data set. “Cartoon Version”:

Step 1. Embed the data set in $\mathbb{R}^{\mathrm{d}}$.

Step 2. Choose a value of σ to define a length scale. Build the data matrix

$$M = (\exp\{ - |x_i - x_j|^2/\sigma\})$$

Step 3. Compute the Eigenvectors $\{\Phi_k\}$.

DG Continued

4. Carefully choose a small number of eigenfunctions, e.g. Φ_3 , Φ_4 , Φ_7 . The new data set representation is given by the image

   $$x_i \rightarrow \{\Phi_3(x_i), \Phi_4(x_i), \Phi_7(x_i)\}$$

5. Why do it? It could be helpful where PCA works poorly. It computes "local affinities" and builds coordinates from that information.

Fast Randomized Algorithms

1. Fast randomized SVD's (V. Rokhlin et al.)
2. A Randomized Approximate Nearest Neighbor Algorithm (P. Jones, A. Osipov, V. Rokhlin)

Why Needed? Building the transition matrix is costly. ($O(N^2)$ for naïve algorithm.)

What these Allow You to Do

- Rapidly build (threshholded) self-adjoint transition matrices using new ANN algorithm
- Use fast SVD’s to compute approx. eigenvectors, apply methods of diffusion geometry.
- This allows rapid updating of the matrix to account for changing scenario.

Data Experiments Demonstrating Potential Applicability

Approach

Represent windows by their product coefficients

Compute low-dimensional models by diffusion mapping

Low-Dimensional Model of Operational Regimes

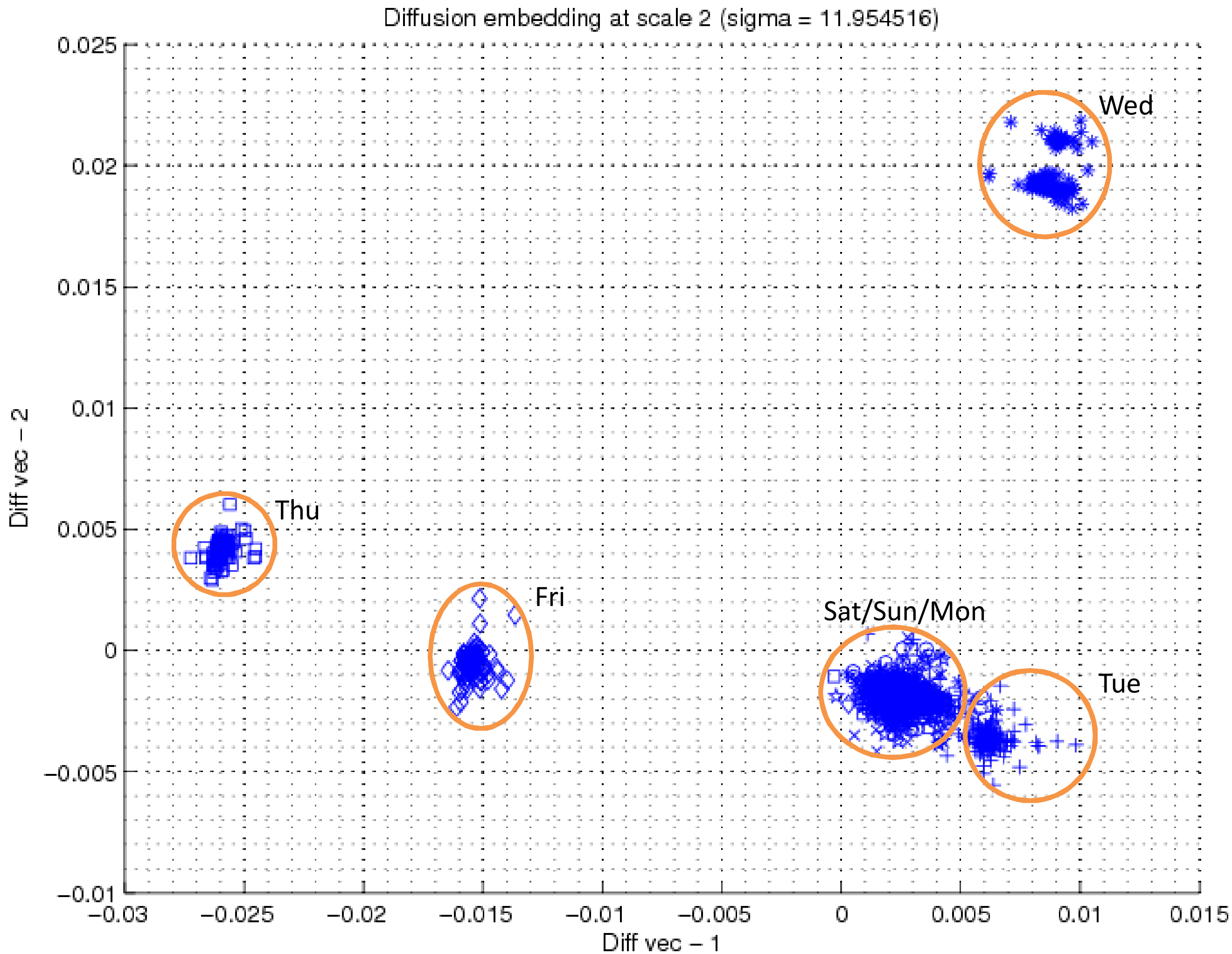
- Daily profile
  - 182 antennas
  - 14 days
  - @ 15 mins
- Analysis

  ~2550 points

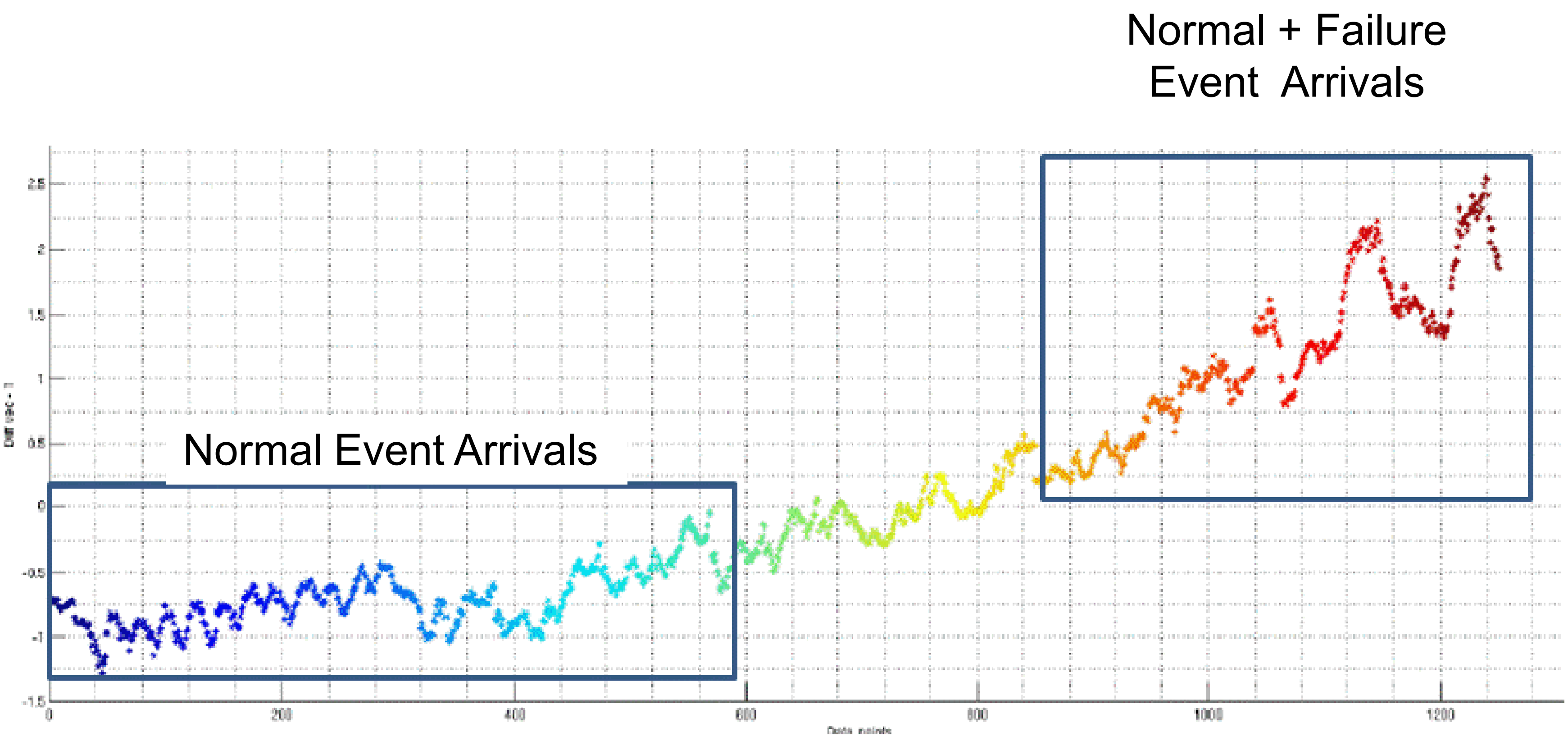
  96 dim.

  a) PDPM

  b) Diffusion Map
- Predict dynamics



Detection of Subtle Regime Change



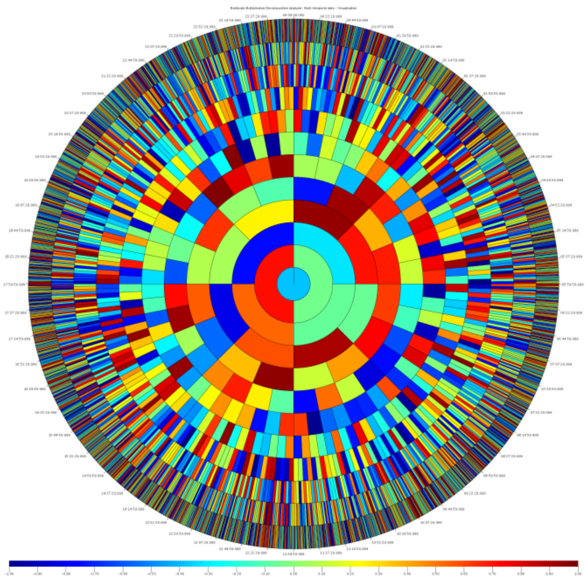
Simulation Scenario proposed by Ron Skoog (Telcordia) to Model Actual Network Cascading Failure

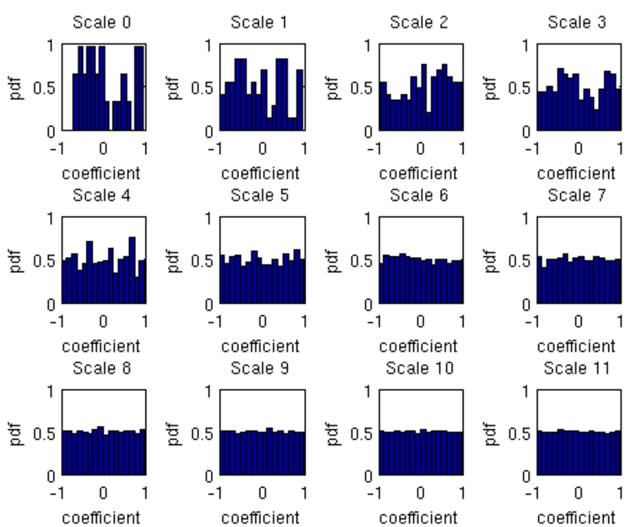
Multi-scale Synthesis

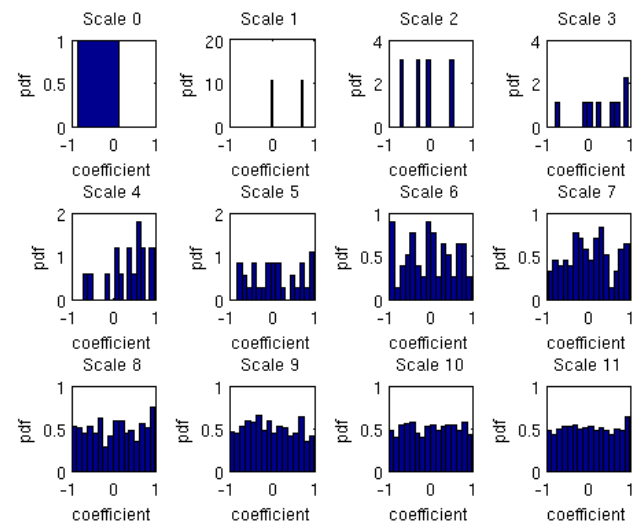
Fix a Probability Density Function on [-1,+1]. Choose the coefficients a_I independently from that PDF. In the following we show a simulated signal, followed by a color chart (using "Jet") of the coefficients.

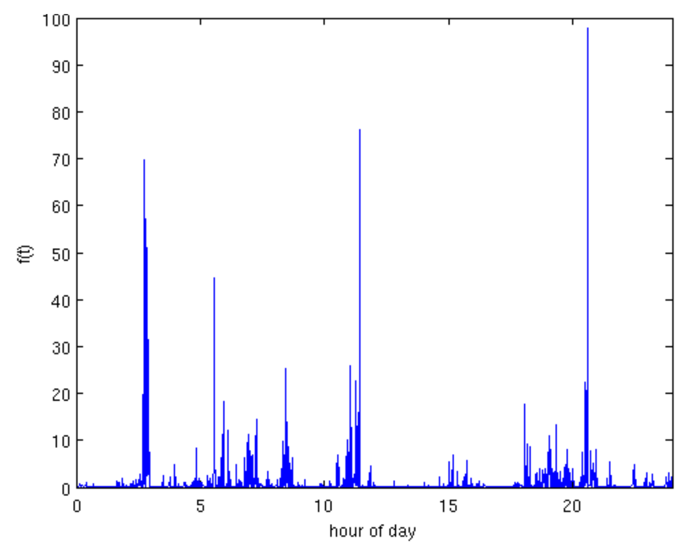
This allows one to simulate for purposes of planning. Similar (but also different!) from other methods (e.g. Barral and Mandelbrot). They need to normalize measure "at every step".

PC Synthesis/Visualization – Uniform

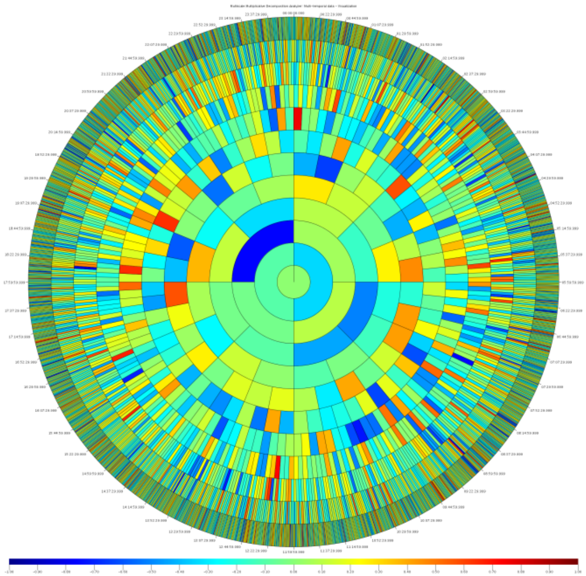


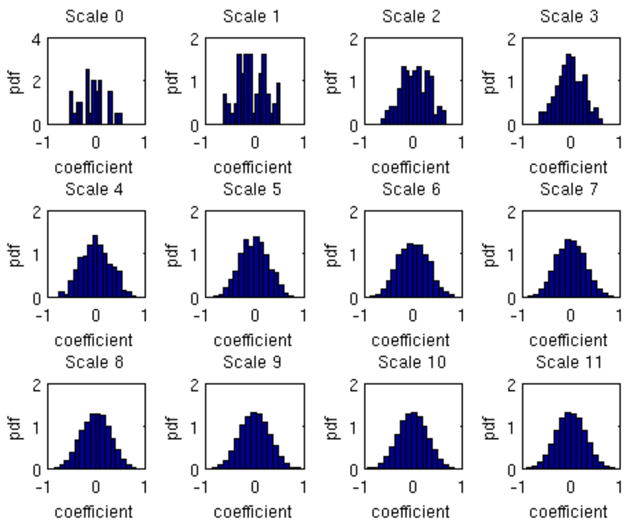


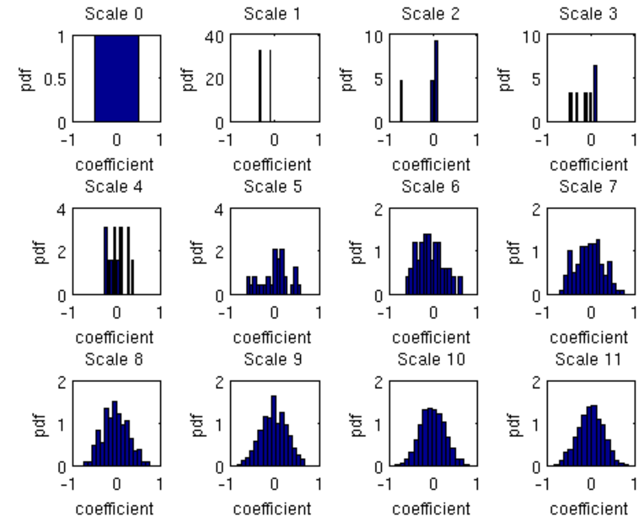


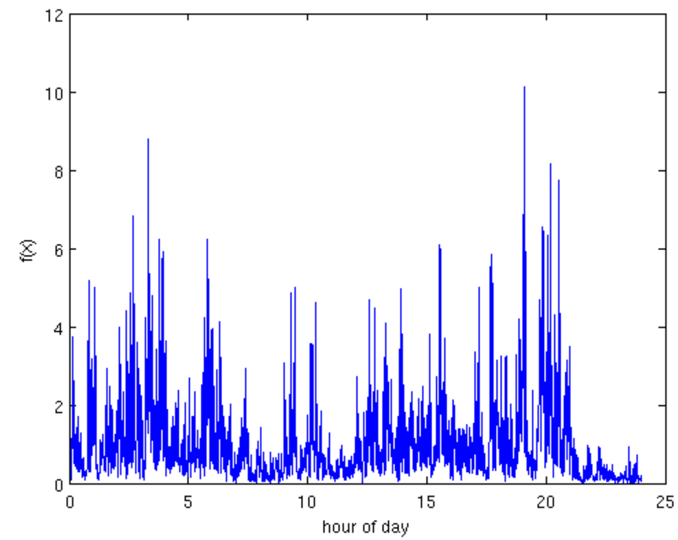


PC Synthesis/Visualization – Bell

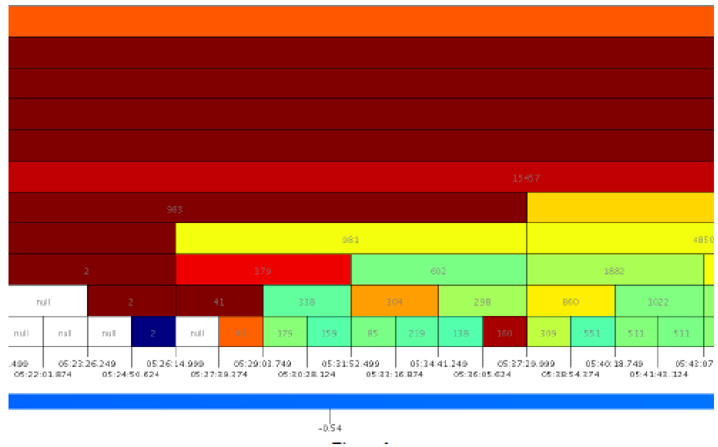


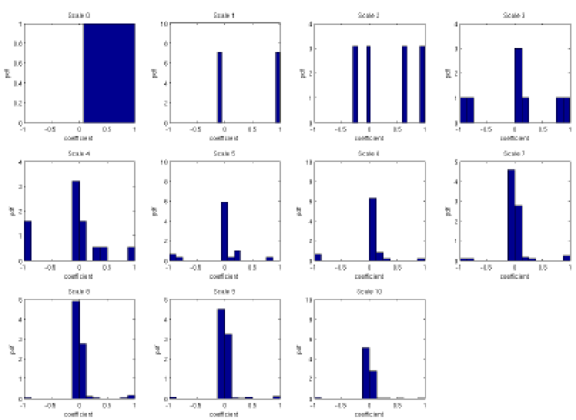






PC Synthesis Model for Network Event





Distribution of product coefficients at various scales

References

History in Analysis

This product formula method and its “cousins” have been around for a while for analysis of various classes of weight functions on Euclidean space. See e.g.

(P. Jones, J. Garnett) *BMO from dyadic BMO.* Pacific J. Math. (1982)

One idea is to use the dyadic setting, analyze functions or weights, and average over translations of the dyadic grid.

Diffusion Geometry References

- For a tutorial see: Mauro Maggioni's Homepage. Click on "Diffusion Geometry".
- Why can it work? See:

"Manifold parametrizations by eigenfunctions of the Laplacian and heat kernels". (P. Jones, Mauro Maggioni and Raanan Schul)
PNAS, vol. 105 no. 6 (2008), pages 1803-1808

(Plus full paper on Raanan Schul's Homepage)

Randomized Algorithms References

1. “Fast SVD’s”, V. Rokhlin et al.
2. “A Randomized Approximate Nearest Neighbor Algorithm”, by P. Jones, A. Osipov, V. Rokhlin.

“Yale CS Technical Report”, (http://www.cs.yale.edu/publications/techreports.html#2010).

Why Needed? Building the transition matrix is costly. ($O(N^2)$ for naïve algorithm.)